

# Datenbanken und Datensicherheit

## Übungsaufgaben 4 – 6. November 2024

### Aufgabe 1: Begrüßungs-Skript

Wir betrachten das Shell-Skript `aufgabe-1.sh`:

```
#!/bin/bash

echo -n "Ihr Name: "
read name
echo "Sei begrüßt, @@@!" | sed -e s/@@@/$name/
```

Das Skript wird zweimal ausgeführt. Danach befindet sich im aktuellen Verzeichnis eine neue Datei `test.txt`:

```
$ ./aufgabe-1.sh
Ihr Name: Peter
Sei begrüßt, Peter!
$ ./aufgabe-1.sh
Ihr Name: mein_Diener!\nI_OWN_U!!!/ -e lwttest.txt -e s/foo/bar
Sei begrüßt, mein_Diener!
I_OWN_U!!!
$ cat test.txt
Sei begrüßt, mein_Diener!
I_OWN_U!!!
```

- (a) Erklären Sie die beabsichtigte Funktionsweise des Skripts (im ersten Aufruf zu sehen).
- (b) Wieso stellt das Verhalten des Skripts im zweiten Aufruf ein Sicherheitsproblem dar?
- (c) Wie ließe sich das Sicherheitsproblem beseitigen?
- (d) Wieso enthält der eingegebene „Name“ Unterstriche anstelle von Leerzeichen?
- (e) Wozu dient das `-e s/foo/bar` am Ende des eingegebenen „Namens“?

**Hinweis:** `man sed`

### Aufgabe 2: Dateitransfer

Auf einem Computer („Server“) geben wir den folgenden Befehl ein:

```
$ nc -p 1234 -l > test.tex
```

Ein anderer Computer („Client“) verbindet sich mit dem Server. (Der Einfachheit halber verwenden wir hier nur einen Computer und die IP-Adresse `127.0.0.1`.)

```
$ nc -q0 127.0.0.1 1234 < dbs-uebung-20241106.tex
```

Danach befindet sich auf dem Server eine neue Datei `test.tex` mit demselben Inhalt wie die Textdatei `dbs-uebung-20241106.tex` auf dem Client. Wir haben also eine Textdatei über das Netz vom Client auf den Server kopiert.

- (a) Wie funktioniert das?
- (b) Funktioniert dies auch für Binärdateien?
- (c) Wozu dient das `-q0`?
- (d) Erweitern Sie die beiden Befehle so, daß die Datei für die Übertragung komprimiert und am Ziel wieder dekomprimiert wird.

**Hinweise:**

- Bei der verwendeten Version von `nc` handelt es sich um `nc.traditional`.
- `man gzip`

*Viel Erfolg!*